

MAVEN ECLISPE IDE INTEGRATION

http://www.tutorialspoint.com/maven/maven_eclipse_ide.htm

Copyright © tutorialspoint.com

Eclipse provides an excellent plugin [m2eclipse](#) which seamlessly integrates Maven and Eclipse together.

Some of features of m2eclipse are listed below

- You can run Maven goals from Eclipse.
- You can view the output of Maven commands inside the Eclipse using its own console.
- You can update maven dependencies with IDE.
- You can Launch Maven builds from within Eclipse.
- It does the dependency management for Eclipse build path based on Maven's pom.xml.
- It resolves Maven dependencies from the Eclipse workspace without installing to local Maven repository (requires dependency project be in same workspace).
- It automatic downloads required dependencies and sources from the remote Maven repositories.
- It provides wizards for creating new Maven projects, pom.xml and to enable Maven support on existing projects
- It provides quick search for dependencies in remote Maven repositories

Installing m2eclipse plugin

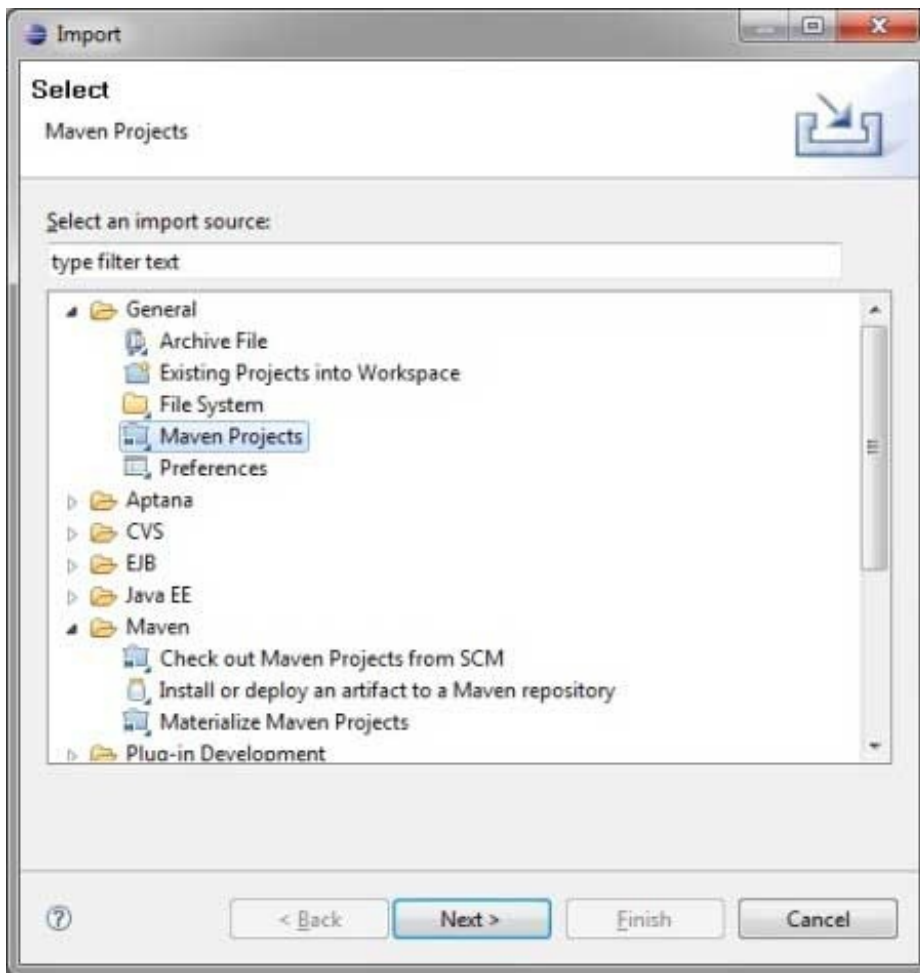
Use one of the following links to install m2eclipse:

Eclipse	URL
Eclipse 3.5 (Gallileo)	Installing m2eclipse in Eclipse 3.5 (Gallileo)
Eclipse 3.6 (Helios)	Installing m2eclipse in Eclipse 3.6 (Helios)

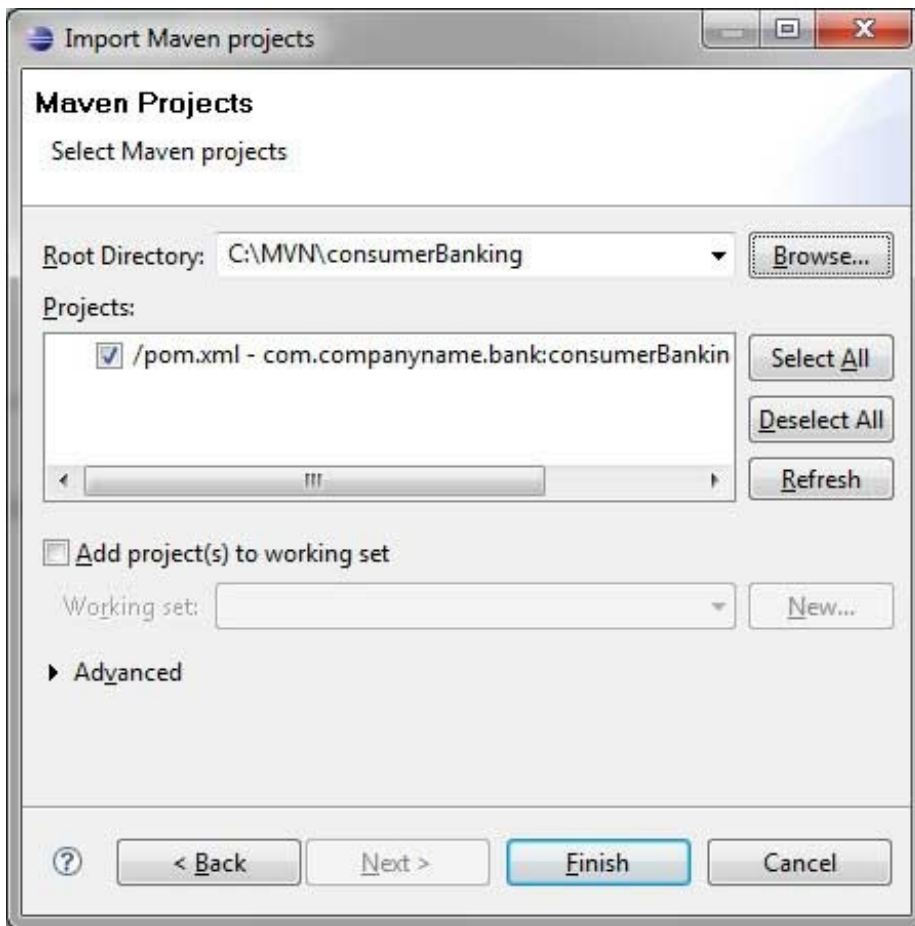
Following example will help you to leverage benefits of integrating Eclipse and maven.

Import a maven project in Eclipse

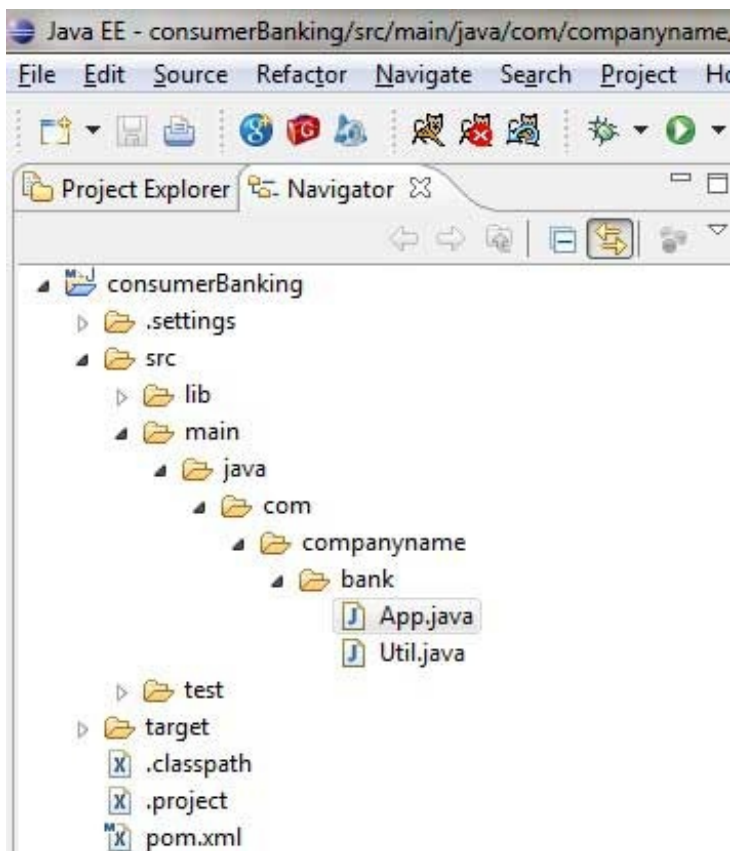
- Open Eclipse.
- Select **File > Import >** option.
- Select Maven Projects Option. Click on Next Button.



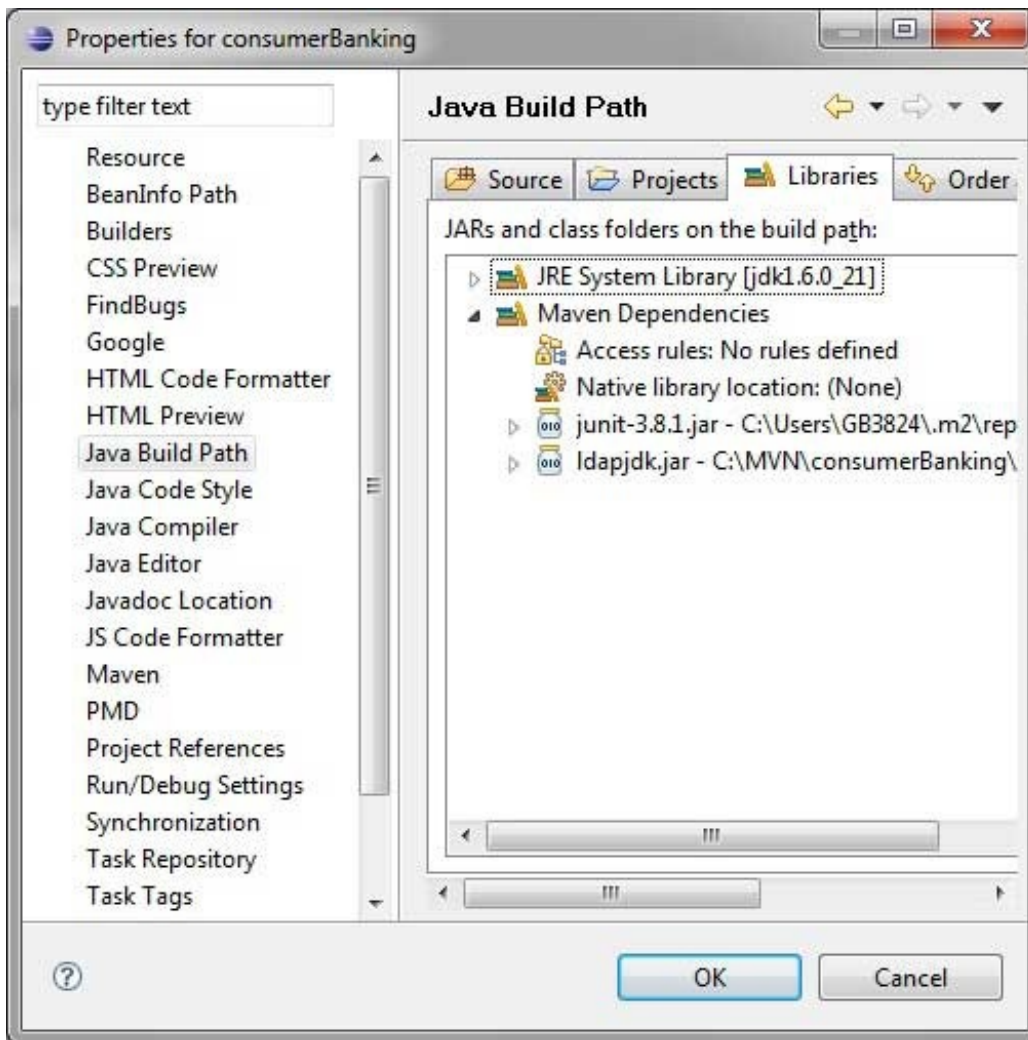
- Select Project location, where a project was created using Maven. We've create a Java Project consumerBanking.See [Maven Creating Project](#) to see how to create a project using Maven.
- Click Finish Button.



Now, you can see the maven project in eclipse.



Now, have a look at consumerBanking project properties. You can see that Eclipse has added Maven dependencies to java build path.



Now, Its time to build this project using maven capability of eclipse.

- Right Click on consumerBanking project to open context menu.
- Select Run as option
- Then maven package option

Maven will start building the project. You can see the output in Eclipse Console

```
[INFO] Scanning for projects...
[INFO] -----
[INFO] Building consumerBanking
[INFO] Id: com.companyname.bank:consumerBanking:jar:1.0-SNAPSHOT
[INFO] task-segment: [package]
[INFO] -----
[INFO] [resources:resources]
[INFO] Using default encoding to copy filtered resources.
[INFO] [compiler:compile]
[INFO] Nothing to compile - all classes are up to date
[INFO] [resources:testResources]
[INFO] Using default encoding to copy filtered resources.
[INFO] [compiler:testCompile]
[INFO] Nothing to compile - all classes are up to date
[INFO] [surefire:test]
[INFO] Surefire report directory:
C:\MVN\consumerBanking\target\surefire-reports

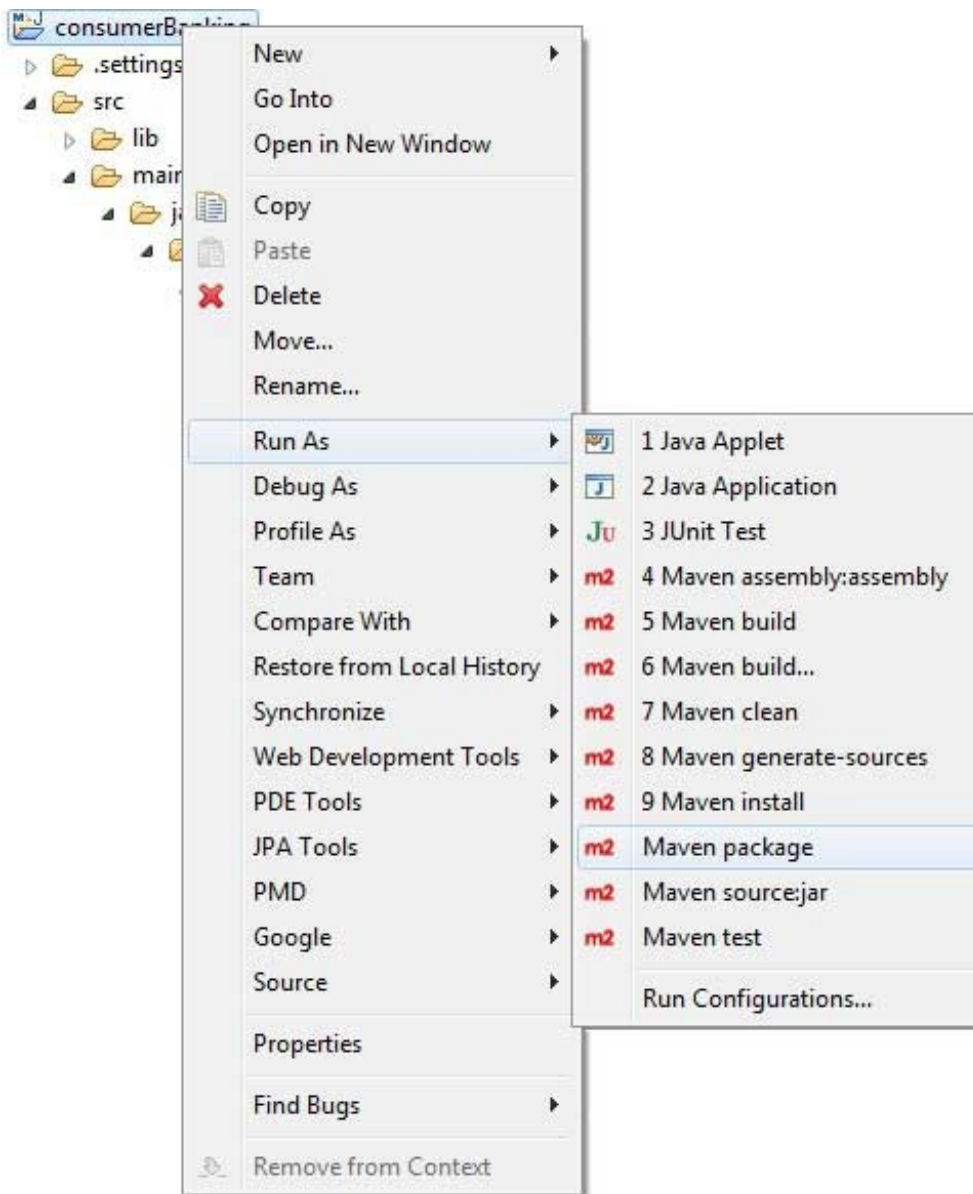
-----
T E S T S
-----
```

```
Running com.companyname.bank.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.047 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] [jar:jar]
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 1 second
[INFO] Finished at: Thu Jul 12 18:18:24 IST 2012
[INFO] Final Memory: 2M/15M
[INFO] -----
```



Now, right click on App.java. Select Run As option. Select As Java Application.

You will see the result

```
Hello World!
```